Copyright © 1971 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 50, No. 5, May-June, 1971
Printed in U.S.A.

On a Class of Rearrangeable Switching Networks

Part II: Enumeration Studies and Fault Diagnosis

By D. C. OPFERMAN and N. T. TSAO–WU

(Manuscript received December 1, 1970)

The decomposition of permutations as used in the control algorithm for a class of rearrangeable switching networks is proved. Enumeration studies on permutations related to the network are presented. Theorems for constructing a set of traffic patterns for diagnostic purposes are also given. Finally, a procedure for detecting and locating faulty switching elements in the network is described.

I. INTRODUCTION

This part of the paper will cover some of the theoretical considerations related to the rearrangeable switching networks discussed in Part I. For the general $(N \times N)$ network with base-d structure, it is shown that it can indeed accommodate any of the $N!$ connection patterns. A thorough study is then made of the $(N \times N)$ network having a base-2 structure. It was pointed out in Part I that the setting of the β-element is, in general, not unique for an arbitrary input-output permutation. Furthermore, the number of β-elements for an $(N \times N)$ network exceeds $\langle \log_2 (N!) \rangle$, for $N > 4$. Some enumeration studies are given to account for this. Finally, fault diagnostic studies are given in relation to the base-2 network. A method to construct a set of permutations useful for testing the network is developed. This is then followed by discussing a procedure to detect and/or locate faulty β-elements in the network.

II. PERMUTATION PROPERTY OF THE NETWORK

In this section it will be shown that the decomposition of the given permutation into reducible connection sets (as used in the control algorithm) is always possible. From Section 3.2.1 of Part I, it is evident that the decomposition is equivalent to the selection of d sets of output integers, $\pi(x_i)$, one from each S_l, such that all the output integers in any of the d sets have distinct characteristics, where S_l, as previously defined, is

$$S_l = \{\pi(x_i) \mid x_i \in J(l, d)\} \qquad 1 \leqq l \leqq N/d;$$

and there are d elements in each S_l .

To show that this selection and, therefore, the decomposition can always be done, P. Hall's Theorem[1] on Distinct Representatives is used and is stated as follows:

P. Hall's Theorem: Let L be a finite set of indices $L = \{1, 2, \cdots n\}$. *For each* $l \in L$, *let* T_l *be a subset of a set T. A necessary and sufficient condition for the existence of distinct representatives* t_l , $l = 1, 2, \cdots, n$, $t_l \in T_l$, $t_i \neq t_j$ *when* $i \neq j$, *is that for every* $k = 1, 2, \cdots, n$ *and every choice of k distinct indices* $l_1, l_2, \cdots, l_k$, *the subsets* $T_{l_1}, T_{l_2}, \cdots, T_{l_k}$ *contain between them at least k distinct elements.*

This theorem can be used directly if a mapping ϕ is defined on the sets S_l as follows:

$$S_l = \{\pi(x_i)\} \xrightarrow{\phi} T_l = \{t\} \qquad 1 \leqq l \leqq N/d,$$

where

$$t = \left[\frac{\pi(x_i) + d - 1}{d}\right]^*.$$

This simply means that each integer in S_l is replaced by its characteristic t, with T_l having exactly the same number of elements as S_l. Thus, the selection of d sets of $\pi(x_i)$, one from each S_l, such that the integers in each of the d sets have distinct characteristics, is equivalent to the selection of d sets of N/d *distinct* representatives, one from each T_l, such that in each of the d sets $t_i \neq t_j$ for $i \neq j$.

By Hall's theorem, it is sufficient to show that for every $k = 1, 2, \cdots, N/d$ and choice of k distinct indices $l_1, l_2, \cdots, l_k$, the sets $T_{l_1}, T_{l_2}, \cdots, T_{l_k}$ contain between them at least k distinct elements. But this is clearly the case here, since each set S_l , and, therefore, each set T_l , contains exactly $(d - j)$ elements after j sets have been so selected. $0 \leqq j \leqq d - 1$. Thus, there are $k(d - j)$ elements in the sets $T_{l_1}, T_{l_2}, \cdots, T_{l_k}$, of which *at most* $(d - j)$ elements are identical (derived from the fact that there are *at most* $(d - j)$ output integers belonging

* $[z]$ is the integral value of z.

to the same integer set after j sets have been so selected). Therefore, there are *at least* k distinct elements. The index j is introduced to show that the selection of d sets of $\pi(x_i)$ can be made on a sequential basis.

III. SOME RESULTS ON ENUMERATIONS

For the remaining sections, the discussion will be restricted to the $(N \times N)$ network with base-2 structure. Some definitions (in addition to those in Part I) relevant to the enumeration study as well as the network diagnosis are given first.

3.1 *Definitions*

(*i*) For any given connection set C, $C \subseteq P$, having input-output pairs (the outputs are denoted as y_i instead of $\pi(x_i)$ to simplify the notations),

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ y_1 & y_2 & \cdots & y_m \end{bmatrix} \qquad 1 \leqq m \leqq N,$$

there exists an inverse of C, denoted by C^{-1}, which is a connection set

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_m \\ x_1 & x_2 & \cdots & x_m \end{bmatrix}.$$

(*ii*) For any two connection sets C_i and C_j that have the same set of input (x_i) and output (y_i) integers, the product $C_iC_j^{-1}$ and its cycle can be defined in the standard manner, similar to that usually associated with permutations.[2]

(*iii*) A loop is a connection set where, for any

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \in L, \quad \begin{bmatrix} x_j = \hat{x}_i \\ y_j \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_k \\ y_k = \hat{y}_i \end{bmatrix} \in L.$$

The number of these input-output pairs in L is called the order of L, which is necessarily even. Moreover, all loops are distinct, i.e., any two loops do not have any common input-output pair.

(*iv*) A proper loop is a loop in which the input-output pairs are arranged so that both x and $\hat{x}$ and y and $\hat{y}$ are adjacent in a circular sense, e.g.,

$$L = \begin{bmatrix} x_1 & x_2 = \hat{x}_1 & x_3 & x_4 = \hat{x}_3 & x_5 & x_6 = \hat{x}_5 \\ y_1 = \hat{y}_6 & y_2 & y_3 = \hat{y}_2 & y_4 & y_5 = \hat{y}_4 & y_6 \end{bmatrix}.$$

This can always be done, and any loop is considered to be a proper loop, unless otherwise specified. Any permutation P on N integers can be written as

$$P = (L_1\,,\,L_2\,,\,\cdots\,,\,L_m)$$

and is said to have m loops, $1 \leqq m \leqq N/2$.

(v) A loop L, of order $2k$, is said to be decomposed into two independent connection sets C_1 and C_2, C_1, $C_2 \subseteq L$, if for any pair $\begin{pmatrix} x_i \\ y_i \end{pmatrix} \in C_1$,

$$\begin{bmatrix} x_j = \hat{x}_i \\ y_j \end{bmatrix} \text{ and } \begin{bmatrix} x_k \\ y_k = \hat{y}_i \end{bmatrix} \in C_2\,.$$

(vi) The derived sets Q_1 and Q_2 (obtained from independent connection sets C_1 and C_2 respectively by replacing every integer by its characteristic) are denoted by $(Q_1\,,\,Q_2)$. If C_1 and C_2 are reducible, Q_1 and Q_2 are permutations P_1 and P_2 respectively, and they are referred to as derived permutations.

3.2 *Enumeration of Permutations by Loops*

In terms of the definitions just given, the looping procedure for the control of the $(N \times N)$ network, described in Section 4.2 of Part I, is equivalent to arranging the given permutation having m loops into the form

$$P = (L_1\,,\,L_2\,,\,\cdots\,,\,L_m) \qquad 1 \leqq m \leqq N/2,$$

and decomposing it to two reducible connection sets C_1 and C_2 by grouping the alternate input-output pairs from each loop into C_1 and the remaining into C_2. Since the decomposition is not unique if $m > 1$, it is readily seen that for any permutation with m loops, there are 2^{m-1} possible ways of decomposition. This leads naturally to the question of how many of the $N!$ permutations have m loops, $1 \leq m \leq N/2$.

The following lemmas and theorems will establish a natural relation between cycles and loops. The enumeration of permutations with m loops $(1 \leqq m \leqq N/2)$ can be expressed in terms of that of cycles, which have been well studied.[3]

Lemma 1: The derived sets $(Q_1\,,\,Q_2)$ *of a loop* L *of order* $2k$ *have the same set of integers* x_i *and* y_i *, and the product* $Q_1Q_2^{-1}$ *has one cycle of length* k.

Proof: Indeed, by definition (v), for every input (or output) integer in C_1 , its dual is in C_2 ; and since they have the same characteristic,

Q_1 and Q_2 have the same set of input (or output) integers. Furthermore, since L is a loop, C_1 and C_2 are of the following form

$$C_1 = \begin{bmatrix} x'_1 & x'_2 & \cdots & x_k \\ \hat{y}'_k & \hat{y}'_1 & \cdots & \hat{y}'_{k-1} \end{bmatrix}, \qquad C_2 = \begin{bmatrix} \hat{x}'_1 & \hat{x}'_2 & \cdots & \hat{x}'_k \\ y'_1 & y'_2 & \cdots & y'_k \end{bmatrix},$$

where $[(x'_i + 1)/2] = [(\hat{x}'_i + 1)/2] = x_i$. Thus, Q_1 and Q_2 are of the form

$$Q_1 = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \\ y_k & y_1 & \cdots & y_{k-1} \end{bmatrix}, \qquad Q_2 = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \\ y_1 & y_2 & \cdots & y_k \end{bmatrix}$$

and, clearly,

$$Q_1 Q_2^{-1} = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \\ x_k & x_1 & \cdots & x_{k-1} \end{bmatrix}$$

and has one cycle of length k.

Q.E.D.

Corollary 1.1: There are 2^{2k-1} loops that give identical derived sets (Q_1, Q_2).

This is clear from the fact that there are $2k$ integers (input and output) in Q_1, and for each (Q_1, Q_2), there are 2^{2k} possible pairs C_1 and C_2. For any given pair C_1 and C_2, the pair C_2 and C_1 reduces to the same (Q_1, Q_2); therefore, 2^{2k} is divided by two.

From definition *(iv)*, any P with m loops can be written as

$$P = (L_1, L_2, \cdots, L_m),$$

where $L_1, L_2, \cdots, L_m$ are disjoint loops. Applying Lemma 1 repeatedly on L_i, the following important theorem that establishes the relation between loops and cycles is obtained.

Theorem 2: The product $P_1 P_2^{-1}$, where P_1 and P_2 are obtained by grouping one Q from each L, has m cycles if and only if P has m loops $(1 \leqq m \leqq N/2)$. As defined in Section 3.1, P_1 and P_2 thus obtained are the derived permutations.

Corollary 2.1: There are 2^{N-m} permutations, having m loops, that will give the same derived permutations (P_1, P_2).

This is proved by repeatedly using Corollary 1.1, and it leads to another enumeration on the number of permutations P that have m loops.

Lemma 3: Define R to be the set of all the derived permutations (P_1, P_2) which have the same product $P_1P_2^{-1}$. Then there are $(N/2)!$ (P_1, P_2) in R.

This is true because both P_1 and P_2 are permutations on $N/2$ integers.

Let $C(n, m)$ denote the number of permutations on n integers that have m cycles. Then there are $C(N/2, m)$ distinct products $P_1P_2^{-1}$ that have m cycles. As a direct consequence of Corollary 2.1 and Lemma 3, the following theorem is established.

Theorem 4. There are exactly $2^{N-m}(N/2)!\ C(N/2, m)$ permutations P which have m loops.

Thus, the enumeration of permutations by loops is related, in a simple manner, to the enumeration of permutations by cycles. The latter problem has been well studied,[3] and the enumeration is generally expressed in terms of the Stirling numbers[4] of the first kind, $s(n, m)$, as follows:

$$C(n, m) = (-1)^{n+m}s(n, m),$$

where $s(n, m)$ can be evaluated from the following generating function

$$\sum_{m=0}^{n} s(n, m)t^m = t(t-1)\cdots(t-n+1)$$

and $(-1)^{n+m}s(n, m)$ is always positive. For the interesting case $m = 1$, the number of permutations with one loop is

$$(N/2)!(N/2-1)!2^{N-1}.$$

3.3 *An Example*

This enumeration is illustrated with the case $N = 8$. If the number of permutations P that have m loops is denoted by $D(N, m)$, Table I

TABLE I—THE NUMBER OF PERMUTATIONS $D(N, m)$ THAT HAVE m LOOPS

m	$C(4, m)$	$D(8, m)$
1	6	$2^7.4!\cdot6 = 18{,}432$
2	11	$2^6.4!\cdot11 = 16{,}896$
3	6	$2^5.4!\cdot6 = 4{,}608$
4	1	$2^4.4!\cdot1 = 384$
		Total $8! = 40{,}320$

accounts for all the permutations. Wherever $m > 1$, there are more than one setting of the β-elements in the input and output stages that will satisfy the same permutation. This applies to all the stages as each subnetwork is taken into consideration, and, thus, the total number of states provided by all the β-elements exceeds $\langle \log_2(N!) \rangle$.

IV. CONSTRUCTION OF TEST PERMUTATIONS

It has been pointed out above that for any P having m loops, certain m β-elements at the input and output stages can be arbitrarily set. To detect faulty β-elements, one must find a class of input-output permutations, or test permutations which are realized by a unique setting of the β-elements. The property of such a permutation is that it and all its derived permutations, at every level of the network, have exactly one loop. To show that they do exist and can be generated, one proceeds as follows:

Lemma 5: If a loop L is given, then any loop L' formed from L by taking the dual of one or more of the integer pairs (input or output) in L will give the same derived sets (Q_1, Q_2), (Q_2, Q_1) being considered the same as (Q_1, Q_2) for the remaining discussion.

This is obvious from the fact that the characteristic of an integer is not changed by taking its dual.

Theorem 6: Let L and L' be two loops having the same (Q_1, Q_2). Then the product $L(L')^{-1}$ has one cycle if and only if L' is obtained from L by replacing every integer except one integer pair (input or output) in L by its dual.

Proof: That L and L' do have the same (Q_1, Q_2) is a direct consequence of Lemma 5. Moreover, the loops L and L' have the same set of x_i and y_i, since x_i, $\hat{x}_i$, y_i, $\hat{y}_i$ are all in L. Thus the product $L(L')^{-1}$ is defined. Now, let L of order $2k$ be written as follows:

$$L = \begin{pmatrix} \hat{x}_1 & x_2 & \hat{x}_2 & \cdots & x_k & \hat{x}_k & x_1 \\ y_1 & \hat{y}_1 & y_2 & \cdots & \hat{y}_{k-1} & y_k & \hat{y}_k \end{pmatrix}$$

and

$$L' = \begin{pmatrix} x_1 & \hat{x}_2 & x_2 & \cdots & \hat{x}_k & x_k & \hat{x}_1 \\ \hat{y}_1 & y_1 & \hat{y}_2 & \cdots & y_{k-1} & y_k & \hat{y}_k \end{pmatrix},$$

where, without loss of generality, the only unchanged pair is y_k and $\hat{y}_k$, since the ordering of subscripts is immaterial. The transformation from L to L' can be expressed in terms of the input-output pairs, namely, $\begin{Bmatrix} \hat{x}_i \\ y_i \end{Bmatrix}$ is replaced by $\begin{Bmatrix} x_i \\ \hat{y}_i \end{Bmatrix}$ and $\begin{Bmatrix} x_i \\ \hat{y}_{i-1} \end{Bmatrix}$ by $\begin{Bmatrix} \hat{x}_i \\ y_{i-1} \end{Bmatrix}$ except $\begin{Bmatrix} \hat{x}_k \\ y_k \end{Bmatrix}$ is replaced by $\begin{Bmatrix} x_k \\ y_k \end{Bmatrix}$ and $\begin{Bmatrix} x_1 \\ \hat{y}_k \end{Bmatrix}$ by $\begin{Bmatrix} \hat{x}_1 \\ \hat{y}_k \end{Bmatrix}$. The input-output pair for $L(L')^{-1}$ will be, in general, $\begin{Bmatrix} \hat{x}_i \\ \hat{x}_{i+1} \end{Bmatrix}$ for $1 \leqq i < k$ and $\begin{Bmatrix} x_i \\ x_{i-1} \end{Bmatrix}$ for $1 < i \leqq k$ with the exception of $\begin{Bmatrix} \hat{x}_k \\ \hat{x}_k \end{Bmatrix}$ and $\begin{Bmatrix} x_1 \\ \hat{x}_1 \end{Bmatrix}$. Therefore,

$$L(L')^{-1} = (\hat{x}_1\hat{x}_2 \cdots \hat{x}_k x_k \cdots x_1),$$

where the product written in the familiar cycle form has one cycle of length $2k$.

To show the converse, it is sufficient to show that the loop L', obtained by either taking the dual of every integer in L or taking the dual of every integer except two or more integer pairs, will not satisfy the second property. Referring to the above, it is seen that if every integer is replaced by its dual, then the product

$$L(L')^{-1} = (\hat{x}_1\hat{x}_2 \cdots \hat{x}_k)(x_k x_{k-1} \cdots x_1)$$

has two cycles, each of length k.

If there are more than two integer-pairs unchanged, one can always write

$$L' = \begin{Bmatrix} x_1 & \hat{x}_2 & x_2 & \cdots & x_j & \hat{x}_{j+1} & \cdots & \hat{x}_k & x_k & \hat{x}_1 \\ \hat{y}_1 & y_1 & \hat{y}_2 & \cdots & y_j & \hat{y}_j & \cdots & y_{k-1} & y_k & \hat{y}_k \end{Bmatrix},$$

where the first other unchanged integer pair is y_j and $\hat{y}_j$, $j < k$. Then, by the same argument given above, the product $L(L')^{-1}$ has at least one cycle of length $2j$, namely,

$$(\hat{x}_1\hat{x}_2 \cdots \hat{x}_j x_j x_{j-1} \cdots x_1) \qquad 2j < 2k. \qquad \text{Q.E.D.}$$

Corollary 6:1: If L is a loop of order k, k ≧ 4, there are k such loops L′ where L and L′ give identical derived sets (Q_1, Q_2) and $L(L')^{-1}$ has one cycle.

This is obvious since there are $k/2$ input integer pairs and $k/2$ output integer pairs. The case $k = 2$ is a degenerate one, since taking the dual of the input pair only yields the same loop as the one obtained by taking the dual of the output pair only. Hence, only one L' is possible.

By repeatedly using the above theorem, one can show the following.

Theorem 7: If $P = (L_1, L_2, \cdots, L_m)$ and its derived permutations are (P_1, P_2), then another permutation P′ will have the same derived permutations and $P(P')^{-1}$ will have m cycles if and only if P′ is ob-

tained by taking the dual of every integer except one integer pair (input or output) in each L_i, $1 \leqq i \leqq m$, in P.

Corollary 7.1: There are $k_1k_2 \cdots k_m$ ways of deriving P' such that P and P' give identical derived permutations (P_1, P_2) and $P(P')^{-1}$ has m cycles, where k_i is the order of L_i and $k_i \geqq 4$. If $k_i = 2$ for some L_i, it will be taken as unity.

The following example illustrates what has been discussed. P has two loops, and the derived permutations (P_1, P_2) have the property that $P_1P_2^{-1}$ has two cycles.

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 8 & 5 & 1 & 9 & 7 & 11 & 3 & 12 & 2 & 10 & 4 & 6 \end{pmatrix}$$

$$= \left(\begin{array}{cc} \begin{matrix} 1 & 5 & 6 & 8 & 7 & 11 & 12 & 2 \end{matrix} & \begin{matrix} 3 & 9 & 10 & 4 \end{matrix} \\ \underbrace{\begin{matrix} 8 & 7 & 11 & 12 & 3 & 4 & 6 & 5 \end{matrix}}_{\text{a loop}} & \underbrace{\begin{matrix} 1 & 2 & 10 & 9 \end{matrix}}_{\text{a loop}} \end{array}\right).$$

The decomposition of P yields:

$$P_1 = \begin{pmatrix} 1 & 3 & 4 & 6 & 2 & 5 \\ 4 & 6 & 2 & 3 & 1 & 5 \end{pmatrix}; \qquad P_2 = \begin{pmatrix} 3 & 4 & 6 & 1 & 5 & 2 \\ 4 & 6 & 2 & 3 & 1 & 5 \end{pmatrix};$$

and $P_1P_2^{-1} = (1\ 3\ 4\ 6)\ (2\ 5)$ has two cycles. P', which gives the same pair (P_1, P_2), is obtained from P, and one of the 32 possibilities is

$$P' = \begin{pmatrix} 2 & 6 & 5 & 7 & 8 & 12 & 11 & 1 & 4 & 10 & 9 & 3 \\ 7 & 8 & 12 & 11 & 4 & 3 & \underline{6} & \underline{5} & \underline{1} & \underline{2} & 9 & 10 \end{pmatrix}$$

(The underlined integers are the unchanged ones in each loop.) Furthermore, $P(P')^{-1} = (1\ 6\ 7\ 12\ 11\ 8\ 5\ 2)\ (3\ 4\ 9\ 10)$.

The permutations for which $m = 1$ can be used in the generation of the test permutations. This is achieved, for any N, by starting with any permutation on 4 integers that has one loop and applying Theorem 7 repeatedly in an iterative manner. One can show that there are

$$2^{(2N-3+(\log_2 N(\log_2 N-3))/2)}$$

such test permutations by repeatedly using Corollary 2.1 and Corollary 7.1, with $m = 1$. The construction of one of these, based on Theorem 7, is illustrated as follows.

In order to clarify the following discussion, test permutations on N

integers and their derived permutations are denoted as $T(N)$ and $(T_1(N/2), T_2(N/2))$ respectively. If it is desired to construct a $T(16)$, then the first step is to select a $T_1(4)$ and generate $T_2(4)$ such that $T_1(4)\,(T_2(4))^{-1}$ has one cycle. There are 16 $T(4)$ that have one loop; one of these is

$$T_1(4) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{pmatrix}.$$

$T_2(4)$ is obtained by taking the dual of every element except one integer pair (by Theorem 7). One of the four choices is

$$T_2(4) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}.$$

Any permutation that decomposes into $T_1(4)$ and $T_2(4)$ can be used for $T_1(8)$; one of the 128 possible permutations (by Corollary 2.1) is

$$T_1(8) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 8 & 5 & 4 & 7 & 6 & 2 \end{pmatrix},$$

where the connection set corresponding to $T_1(4)$ is taken as

$$\begin{bmatrix} 1 & 3 & 5 & 7 \\ 1 & 8 & 4 & 6 \end{bmatrix}.$$

There are eight choices for $T_2(8)$, and one of these is

$$T_2(8) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 6 & 7 & 8 & 3 & 1 & 5 \end{pmatrix}.$$

Similarly, one of the possible $T(16)$'s is

$$T(16) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 6 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 1 & 4 & 5 & 8 & 12 & 16 & 13 & 10 & 15 & 7 & 6 & 14 & 2 & 11 & 9 & 3 \end{pmatrix}.$$

It will now be shown that a permutation that is realized by complementing every setting of the β-elements that realizes $T(N)$ except the one corresponding to inputs 1 and 2 (see network structure) is also a test permutation, $T^c(N)$, and that it can be generated parallel to $T(N)$. These two permutations are used for fault detection in a manner to be described later.

Theorem 8: *Let $T_1(N)$ be a test permutation that has $(T_1(N/2), T_2(N/2))$ as the derived permutations. And if there exist two other test permutations*

$T_1^c(N/2)$ and $T_2^c(N/2)$ such that the product $T_1^c(N/2)(T_2^c(N/2))^{-1}$ has one cycle, then one can construct a test permutation which will have $(T_1^c(N/2), T_2^c(N/2))$ as the derived permutations.

Proof: Let $T_1(N)$ be decomposed into two reducible connection sets, C_1 and C_2, where C_1 can be written as

$$C_1 = \begin{bmatrix} x_1 = 1 & x_2 & \cdots & x_{N/2} \\ y_1 & y_2 & \cdots & y_{N/2} \end{bmatrix}$$

($x_1 = 1$ is arbitrarily defined by the network structure).

The connection set C_1^c is formed by replacing each input and output integer $z \in T_1^c(N/2)$, except $x_1 = 1$, by x (or y) that has the characteristic z and has its dual $\hat{x} = x_i$, for some i, belonging to C_1, $1 < i \leqq N/2$ (or $\hat{y} = y_i \in C_1$, $1 \leqq i \leqq N/2$). Clearly, this can always be done because every $z \in T_1^c(N/2)$ has two integers with z as their characteristic, and only one of them is in C_1. Similarly, a connection set C_2^c can be formed from $T_2^c(N/2)$, based on C_2. The permutation, obtained simply by combining C_1^c and C_2^c, has the derived permutations $(T_1^c(N/2), T_2^c(N/2))$, and it has only one loop. Furthermore, it results in the complementary setting of β-elements by the looping algorithm, since any integer in C_1^c is the dual of some integer in C_1. Q.E.D.

Theorem 9: One can construct a test permutation $T_2^c(N)$ from $T_1^c(N)$ in the same way as $T_2(N)$ is obtained from $T_1(N)$ as given in Theorem 7, and the product has also one cycle.

Proof: Let $T_2(N)$ be obtained from $T_1(N)$ by taking the dual of every integer except one pair, say, $(x_i, \hat{x}_i)$. The permutation, obtained from $T_1^c(N)$ by taking the dual of every integer pair except the same pair $(x_i, \hat{x}_i)$, is indeed a test permutation by Theorem 7. Using the same argument as in Theorem 8, it is easily seen that the setting of β-element to realize this permutation is complementary to that for $T_2(N)$. Hence it is $T_2^c(N)$. Q.E.D.

Since the permutations $T_1(2)$ and $T_2(2)$, and the corresponding $T_1^c(2)$ and $T_2^c(2)$, can always be constructed, one can, by induction on N, construct the two test permutations $T(N)$ and $T^c(N)$ for arbitrary value of N. One can, in fact, generalize Theorems 8 and 9 to establish arbitrary relations between two permutations in the β-element settings in addition to $T(N)$ and $T^c(N)$ for which the settings are complementary.

The construction procedure for $T^c(N)$ is illustrated by determining the $T^c(16)$ as related to $T(16)$ given in the previous example. In that example, $C_1 = \begin{bmatrix} 1 & 4 \\ 1 & 3 \end{bmatrix}$ and $C_2 = \begin{bmatrix} 2 & 3 \\ 4 & 2 \end{bmatrix}$. Also $T_1(2) = \binom{1\ 2}{1\ 2}$ and $T_2(2) = \binom{1\ 2}{2\ 1}$;

therefore, $T_1^c(2) = \binom{1\ 2}{2\ 1}$ and $T_2^c(2) = \binom{1\ 2}{1\ 2}$. All elements except input 1 in C_1^c must have their dual in C_1 ; also $T_1^c(2)$ and $T_2^c(2)$ must be satisfied; therefore,

$$C_1^c = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}, \quad \text{and} \quad C_2^c = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$$

and

$$T_1^c(4) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}.$$

Keeping the input integer pair (1, 2) unchanged, one has

$$T_2^c(4) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}.$$

Repeating the procedure, one obtains

$$T_1^c(8) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 6 & 4 & 2 & 8 & 3 & 1 & 5 \end{pmatrix},$$

$$T_2^c(8) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 5 & 1 & 3 & 4 & 7 & 6 & 2 \end{pmatrix},$$

and

$$T^c(16) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 13 & 16 & 10 & 12 & 8 & 1 & 4 & 5 & 15 & 7 & 6 & 14 & 2 & 11 & 9 & 3 \end{pmatrix}.$$

V. INVERSE CONNECTING EQUATIONS FOR BASE-2 STRUCTURE

The looping procedure for setting the β-elements to realize a given P is described in Part I. The inverse problem of defining P from the states of the β-elements is also of some interest. If P can be derived from the β-element setting, then it is not necessary to store the connections in another memory. Also the inverse connecting equations are used in the location of faulty β-elements.

In the control algorithm for the base-2 structure, the states of the β-elements are derived in an iterative manner from the outside (first) level to the center $(\log_2 N)^{\text{th}}$ level. Therefore, to obtain the inverse connecting equations, the states of β-elements in the center stage are considered first.

The β-elements in the network are numbered (see Fig. 1) such that

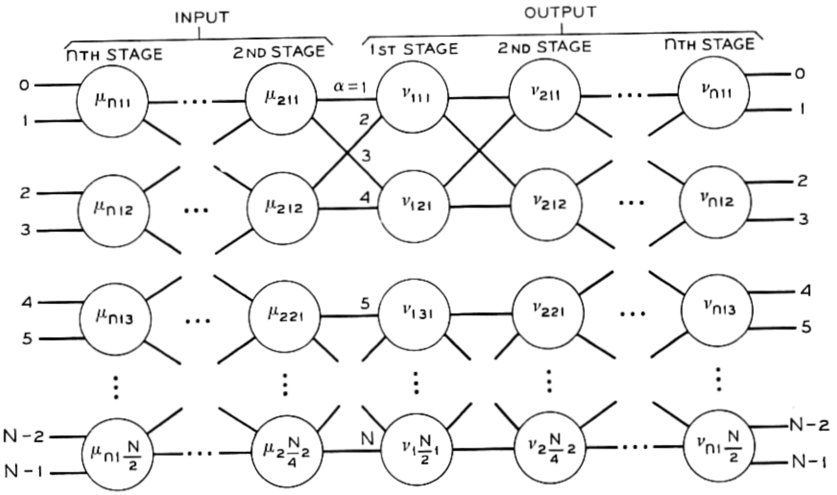

Fig. 1—The numbering of β-elements in an $(N \times N)$ network.

the defining equations for each input-output pair appear in a simple form. μ_{jkl} and ν_{jkl} are the input and output β-elements respectively. They are located in the jth stage (counting from the center stage), the kth $(2^j \times 2^j)$ network, and the lth position at the input (or output) stage. The center stage is considered as the first output stage, and the β-elements are denoted by ν_{1k1}. They are defined as '0' or '1' when set to straight-through state or crossover state, respectively.

The code for the input and output integers is the same as given in Section 4.1 of Part I. For any input integer x_i (or output integer y_i), the normal binary representation is its coded form, having a code length of $n = \log_2 N$. And it is expressed as follows:

$$x_i = x_{i1}x_{i2} \cdots x_{in} .$$

The inputs to the center stage are designated by α, $\alpha = 1, 2, \cdots, N$, as shown in Fig. 1, and the input-output pairs are ordered according to α. For each input-output pair $\binom{x_\alpha}{y_\alpha}$, the code words for $x_\alpha = x_{\alpha 1}x_{\alpha 2} \cdots x_{\alpha n}$ and $y_\alpha = y_{\alpha 1}y_{\alpha 2} \cdots y_{\alpha n}$ can be calculated from the following inverse connecting equations:

$$x_{\alpha 1} = (\alpha + 1) \bmod 2,$$

$$x_{\alpha j} = \mu_{jkl_x}^{\rho \bmod 2*} \qquad 1 < j \leqq n;$$

[*] $z^1 \equiv z$ and $z^0 \equiv \bar{z}$, the complement of z, and $z = 0$ or 1.

and

$$
\begin{aligned}
y_{\alpha 1} &= \nu^{\alpha \bmod 2}_{1[(\alpha+1)/2]1}\ , \\
y_{\alpha j} &= \nu^{\rho \bmod 2}_{jkl_y} \qquad 1 < j \leqq n;
\end{aligned}
\tag{1}
$$

where

$$
\rho = \left[\frac{\alpha - 1 + 2^{i-1}}{2^{i-1}}\right], \qquad k = \left[\frac{\rho + 1}{2}\right]
$$

and l_x and l_y are the integers represented in the coded form by $x_{\alpha 1}x_{\alpha 2} \cdots x_{\alpha(j-1)}$ and $y_{\alpha 1}y_{\alpha 2} \cdots y_{\alpha(j-1)}$ respectively. The equations for $x_{\alpha j}$ (or $y_{\alpha j}$), $1 < j \leqq n$, can be obtained in a recursive manner from the following Boolean equations:

If α and j are such that ρ is even,

$$
x_{\alpha j} = (\bar{x}_{\alpha 1}\bar{x}_{\alpha 2} \cdots \bar{x}_{\alpha(j-1)})\mu_{jk1} + (\bar{x}_{\alpha 1}\bar{x}_{\alpha 2} \cdots x_{\alpha(j-1)})\mu_{jk2} + \cdots
+ (x_{\alpha 1}x_{\alpha 2} \cdots x_{\alpha(j-1)})\mu_{jk(2^{i-1})}\ .
$$

And, if ρ is odd,

$$
x_{\alpha j} = (\bar{x}_{\alpha 1}\bar{x}_{\alpha 2} \cdots \bar{x}_{\alpha(j-1)})\bar{\mu}_{jk1} + (\bar{x}_{\alpha 1}\bar{x}_{\alpha 2} \cdots x_{\alpha(j-1)})\bar{\mu}_{jk2} + \cdots
+ (x_{\alpha 1}x_{\alpha 2} \cdots x_{\alpha(j-1)})\bar{\mu}_{jk(2^{i-1})}\ .
$$

The following example of the (8 × 8) network shown in Fig. 2 illustrates the inverse procedure. For each $\alpha = 1, 2, \cdots, 8$, the coded

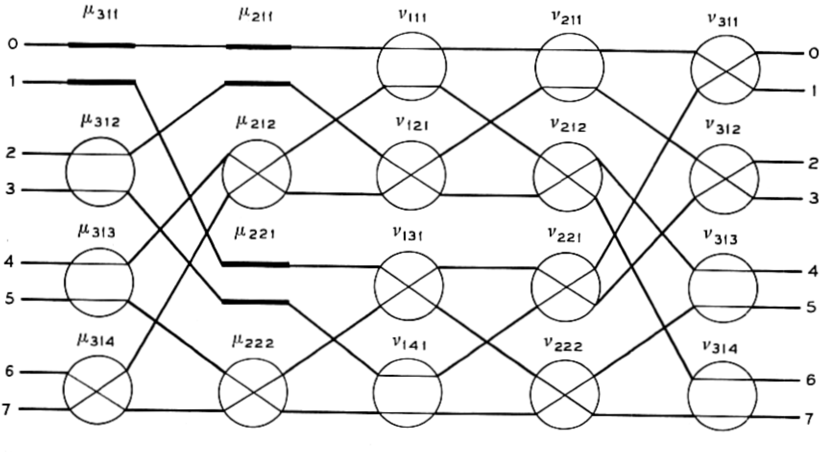

Fig. 2—β-element setting for $P = \binom{01234567}{17403526}$.

inputs and outputs are calculated. If $\alpha = 5$, then

$$x_{51} = (5 + 1) \bmod 2 = 0;$$

$$x_{52} = \mu^{3 \bmod 2}_{2[(3+1)/2]1} = \mu_{221} = 0;$$

and

$$x_{53} = \mu^{2 \bmod 2}_{3[(2+1)/2]1} = \bar{\mu}_{311} = 1.$$

Also

$$y_{51} = \nu^{5 \bmod 2}_{1[(5+1)/2]1} = \nu_{131} = 1;$$

$$y_{52} = \nu^{3 \bmod 2}_{2[(3+1)/2]2} = \nu_{222} = 1;$$

and

$$y_{53} = \nu^{2 \bmod 2}_{3[(2+1)/2]4} = \bar{\nu}_{314} = 1.$$

Therefore, the input-output pair

$$\begin{bmatrix} x_5 \\ y_5 \end{bmatrix} = \begin{bmatrix} 001 \\ 111 \end{bmatrix} = \begin{bmatrix} 1 \\ 7 \end{bmatrix}.$$

The remaining

$$\begin{bmatrix} x_\alpha \\ y_\alpha \end{bmatrix} (\alpha = 1, 2, 3, 4, 6, 7, 8)$$

are determined in the same manner, and they are:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 000 \\ 001 \end{bmatrix}; \quad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 111 \\ 110 \end{bmatrix}; \quad \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} 010 \\ 100 \end{bmatrix}; \quad \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} 100 \\ 011 \end{bmatrix};$$

$$\begin{bmatrix} x_6 \\ y_6 \end{bmatrix} = \begin{bmatrix} 110 \\ 010 \end{bmatrix}; \quad \begin{bmatrix} x_7 \\ y_7 \end{bmatrix} = \begin{bmatrix} 011 \\ 000 \end{bmatrix}; \text{ and } \begin{bmatrix} x_8 \\ y_8 \end{bmatrix} = \begin{bmatrix} 101 \\ 101 \end{bmatrix}.$$

Then the input-output permutation is

$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 7 & 4 & 0 & 3 & 5 & 2 & 6 \end{pmatrix}.$$

VI. DIAGNOSIS OF FAULTY β-ELEMENTS

The physical design of the β-element is the major factor in determining the method of detecting and locating the faulty elements in the network. For example, the detection of a faulty β-element which either opens or shorts to ground is trivial. If one has access to the actual state of each β-element, then the location of faulty elements is also trivial. In this paper it is assumed that the individual β-element is not accessible, and it is considered to fail when it remains in one of the two states.

6.1 *Detection*

By using the test permutations $T(N)$ and $T^c(N)$, each β-element is checked for the two possible states. Failure in any number of β-elements will be detected by the fact that either $T(N)$ or $T^c(N)$ or both will not be realized. It is to be noted that any permutation having more than one loop cannot be used because the setting of some two or more β-elements is arbitrary, and, therefore, failure of these elements in certain states may not be detected.

6.2 *Location*

With any test permutation $T(N)$, failure of one β-element will result in a permutation different from $T(N)$ by only two input-output pairs, that is, the input-output pairs $\binom{x_i}{y_i}$ and $\binom{x_j}{y_j}$ become $\binom{x_i}{y_j}$ and $\binom{x_j}{y_i}$ for some i and j. The inverse connecting algorithm discussed in Section V can be used to locate the particular (or the faulty) β-element common to $\binom{x_i}{y_i}$ and $\binom{x_j}{y_j}$ with their associated α's which are stored in the memory.

The following example illustrates this procedure. If the test permutation for an (8×8) network is

$$T(8) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 7 & 4 & 3 & 6 & 5 & 1 \end{pmatrix},$$

then, using the same coding scheme as in Section V, the setting of the β-elements for it is shown in Fig. 3. Also, the α's corresponding to each input-output pair are calculated by using the inversing connecting equations, and they are given as follows: α_1 , α_2 , α_3 , α_4 , α_5 α_6 , α_7 , and and α_8 correspond to input-output pairs

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 5 \end{bmatrix}, \begin{bmatrix} 2 \\ 7 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \text{ and } \begin{bmatrix} 7 \\ 1 \end{bmatrix} \text{ respectively.}$$

Assume that β-element ν_{212} is faulty, and it is fixed in the crossover position. Then the actual permutation realized is

$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 5 & 4 & 3 & 6 & 7 & 1 \end{pmatrix},$$

and the incorrect pairs of $T(8)$ are $\binom{2}{7}$ and $\binom{6}{5}$ or in coded form $\binom{010}{111}$ and $\binom{110}{101}$. The α's for these pairs are 3 and 2 respectively. By using

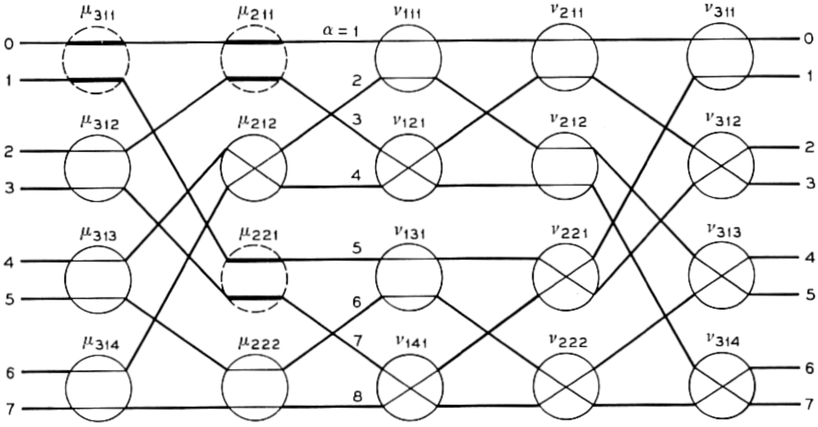

Fig. 3—β-element setting for $T(8) = \binom{01234567}{02743651}$.

the inverse connecting equations (1), the β-elements through which $\binom{2}{7}$ and $\binom{6}{5}$ are connected are found to be μ_{312}, μ_{211}, ν_{121}, ν_{212}, and ν_{314}, and μ_{212}, ν_{111}, ν_{212}, and ν_{313} respectively. It is seen that β-element ν_{212} is common to both input-output pairs; therefore, it is the faulty β-element.

If two β-elements are faulty, there are either three or four pairs in $T(N)$ not realized. If four input-output pairs are wrong, the locations of faulty elements can be determined in the same manner as described above. Three pairs are incorrect when one particular $\binom{x_i}{\nu_i}$ is connected through both of the faulty elements. For this case, it is necessary to change $T(N)$ so that one of the faulty elements is in the proper state, and then the other one can be located. This is achieved by having $2 \log_2 N - 1$ test permutations with each one constructed (using a generalized form of Theorems 8 and 9) to complement different stages of input or output β-elements, one stage at a time. The same procedure as above is used to locate the faulty β-element.

If the faulty β-elements are restricted to one stage of the network, then this stage can be located in a manner similar to the above. For this case, a set of $T(N)$, $\log_2 N$ in number, is used to complement the β-elements on each stage (input and output) of the network. The α's corresponding to the incorrect $\binom{x_i}{\nu_i}$'s will remain the same until the faulty elements are complemented. Therefore, the stage containing the faulty β-elements is determined.

6.3 *Adaptive "Looping" Algorithm*

In the looping algorithm as given in Part I, the derived permutations P_1 and P_2 are always routed through the upper and lower

$(N/2 \times N/2)$ networks respectively. This is because in the most efficient network structure (see Fig. 1 of Part I) one β-element in the first stage of each subnetwork (i.e., μ_{jk1} , $1 < j \leqq \log_2 N$, $1 \leqq k \leqq N/2^j$) is fixed in the straight-through state. However, by introducing redundant β-elements μ_{jk1} , (as in Fig. 1) one can change (or adapt) the control algorithm at certain stages to realize a particular permutation if there is one faulty β-element per subnetwork at each stage.

VII. CONCLUSION

Important relationships between the loops of input-output permutations and cycles of permutations are established. These properties are used to enumerate the input-output permutations in terms of loops and to construct special test permutations which require unique β-element settings. Also, inverse connecting equations which define the input-output permutation from the states of the β-elements are derived. These ideas are utilized in the diagnosis of faulty β-elements.

It is clear that network failure due to any number of faulty elements, which may be distributed over many stages, can be easily detected by using only a pair of test permutations. If these faulty elements are limited to only one stage of the network, this stage can be located by employing the inverse equations and a set of test permutations. Furthermore, if only one or two elements fail, their exact positions in the network can be located by employing a similar procedure.

If the faulty elements are limited to only one in the first stage of each subnetwork, then any input-output permutation can be realized correctly by adding a redundant β-element in the first stage of each subnetwork and adapting the looping algorithm at the appropriate subnetworks.

The fact that this type of rearrangeable switching network has some attractive diagnostic properties should enhance the possibility of it being used in some practical switching systems.

REFERENCES

1. Hall, P., Jr., *Combinatorial Theory,* Waltham, Mass.: Blaisdell Publishing Company, 1967, Chapter 5.
2. Ledermann, W., *Introduction to the Theory of Finite Groups,* New York: Interscience Publishers, 1964, Chapter 3.
3. Riordan, J., *Introduction to Combinatorial Analysis,* New York: John Wiley and Sons, 1958, Chapter 4.
4. National Bureau of Standards, *Handbook of Mathematical Functions,* AMS 55, Washington: National Bureau of Standards, March 1965, p. 824.